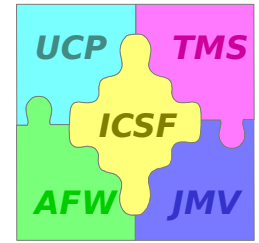


External Database Integration – Garlic Fries

Steve LoCicero



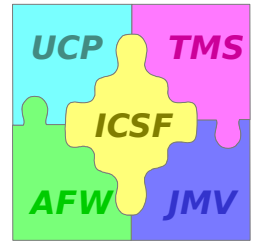
Agenda



- ❑ Overview
- ❑ Architecture
- ❑ APIs
- ❑ Examples



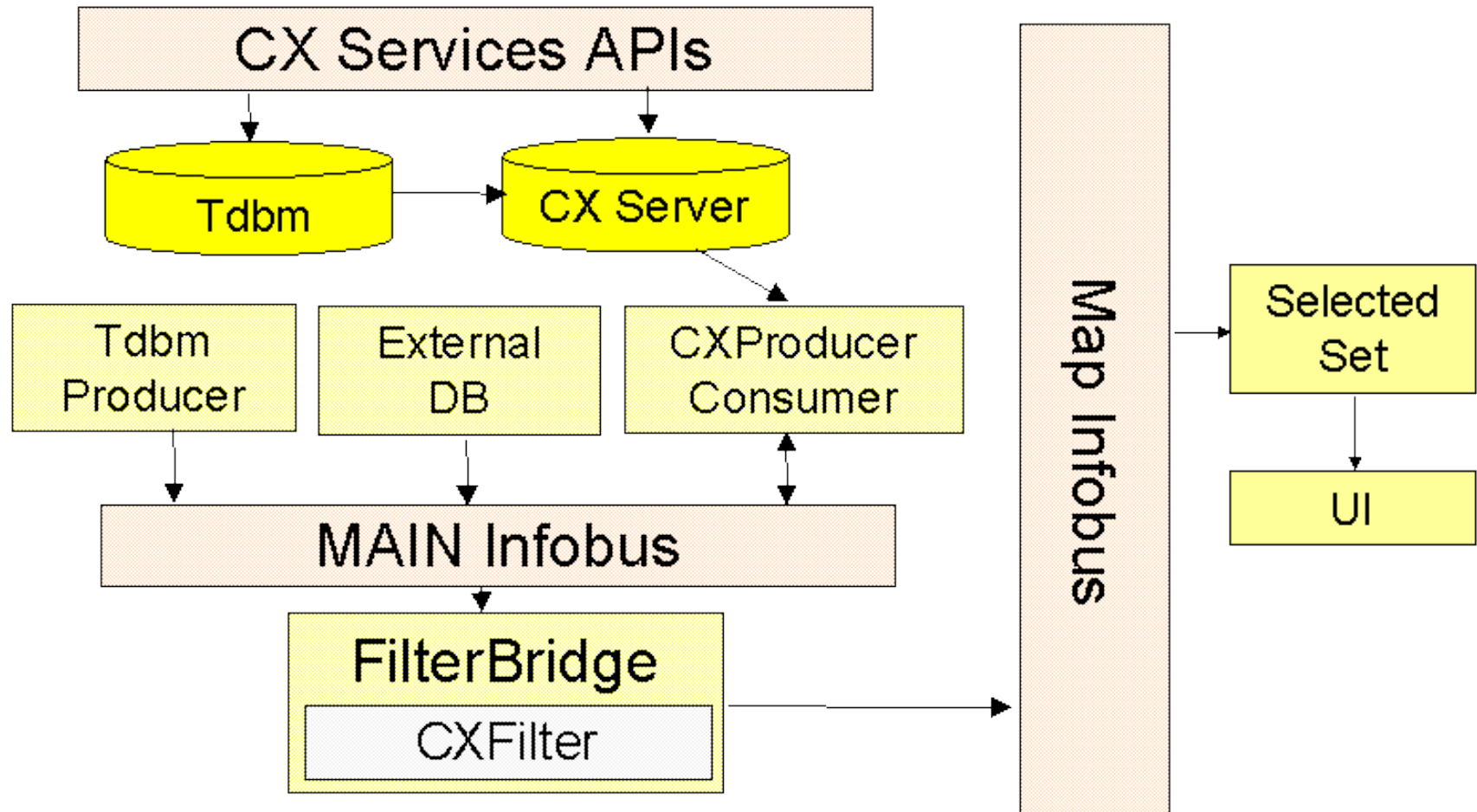
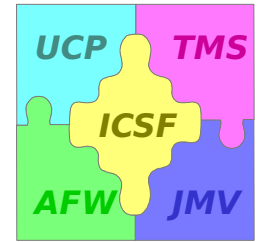
Garlic Fries Overview

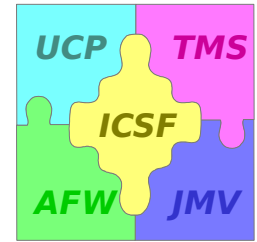


- ❑ *TMS now provides a Cross Reference (CX) service*
- ❑ *CX Services allows tracks to be updated from external sources*
 - *The external sources (CX sources) register with TMS CX Services*
 - *A CX source specific UUID is assigned by the server*
 - *Based on data external to TMS, CX sources (external db managers) submit track updates to TMS*
 - *TMS correlates the updates and either*
 - *Creates a new track or*
 - *Updates an existing track in the track database*
 - *TMS maintains a cross reference from the TMS track with the object stored in the external database.*
- ❑ *Objects from the external database can be plotted and viewed on the display.*
 - *External database objects that are cross referenced to a TMS track appear as one symbol on the display.*



CX Services Architecture



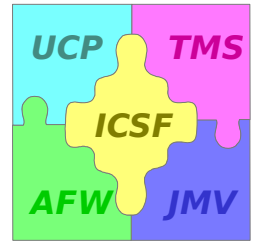


CX Server Tasks

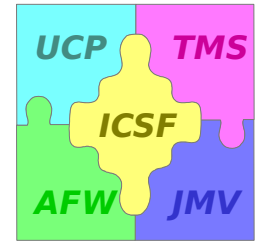
- ❑ *Processes source registration, track injections, and Tdbm events*
- ❑ *Updates cross reference information with Tdbm deletes and merges*
- ❑ *Distributes track cross reference information to TmsCXTable*
- ❑ *Stores source registration information and track cross references*



Producer Consumer Tasks



- ❑ *CX Producer Consumer:*
 - *Produces TmsCXTable on the MAIN Infobus*
 - *Consumes tracks that are produced on the MAIN Infobus*
 - *Handles updates to TmsCXTable*
 - *Additions to the track cross reference*
 - *Deletions to the track cross reference*
 - *Merges to the track cross reference*
- ❑ *External DB Producer:*
 - *Implements **IAfwCXIdentity** on each cross referenced object*
 - *Places each object on the MAIN Infobus.*

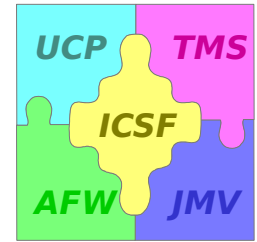


CX Filter Tasks

- ❑ *Ensures tracks that are cross referenced to a TMS track appear as one symbol on the display*
 - *Determines CX identity when IAfwCXIdentity is implemented*
- ❑ *To be filtered, each item must implement the IAfwCXIdentity interface so that the filter can get the item's CX identity.*
 - *Checks the CX table (TmsCXTable) for a parent*
 - *When parent is found, filters out the child (CX) items.*
- ❑ *The Map Infobus will contain only those items that:*
 - *Have not implemented IAfwCXIdentity or*
 - *Do not have parent CX table entry.*



CX Java APIs



Object Actions

- | | |
|---------------|---|
| updateCXTrack | - Injects a track in Tdbm and creates a cross reference entry. |
| deleteTrkCX | - Deletes the track cross reference object associated with the specified track UUID from the cross - reference table. |

Object Creation

- | | |
|----------|---|
| TmsTrkCX | - Creates a new track cross reference object. |
|----------|---|

Object Accessors

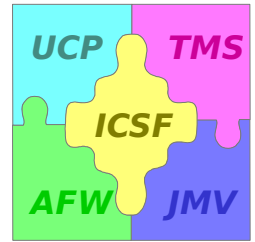
- | | |
|------------|--|
| getSrcUUID | - Gets the source UUID associated with a given track cross reference object. |
| getKey | - Gets the key associated with a given track cross reference object. |

Object Mutators

- | | |
|------------|--|
| setSrcUUID | - Sets the source UUID for a track cross reference object. |
| setKey | - Sets the key for a track cross reference object. |



CX Source Registration



```
import disa.tms.tdbmadapter.*;

class Sample {
    static public void main(String args[]) {

        // Create new VtSrcCX
        TmsSrcCX cx = new TmsSrcCX();

        // Set Information in Source CX Object
        cx.setDb("Tdbm");
        cx.setHost("dii70");
        cx.setTable("UnitTracks");
        cx.setAccess("password");
        cx.setRelationship(2);

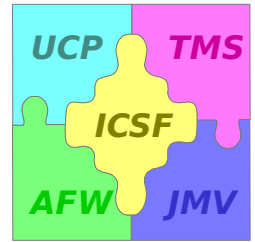
        // Register Source CX with CX Services
        String srcUuid = cx.updateSrcCX();

        // Verify Registration
        if (srcUuid != null)
            System.out.println("src uuid  = " + srcUuid);

    }
}
```



CX Track Injection



```
import disa.tms.tdbmadapter.*;

class Sample {
    static public void main(String args[]) {

        TmsTrkCX cx = new TmsTrkCX();    // Create new VtTrkCX

        // Set Information in Track CX Object
        cx.setSrcUUID("USE_UUID_FROM_SAMPLE_SOURCE_REGISTRATION");
        cx.setKey("123456KEY");

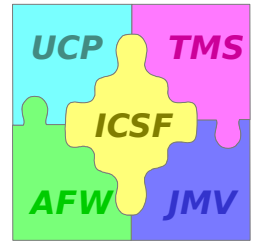
        TmsContext tms_ctxt = new TmsContext();

        // Create Track using TMS APIs
        ITmsTrack trk = new TmsTrackUnit();
        ITmsReport rpt = trk.getReport();
        ITmsPosit e_pos = rpt.getPosit();
        Date dtg = new Date(2000, 5, 24);
        e_pos.setDtg(dtg.getTime()/1000); e_pos.setLat(-50.0); e_pos.setLng(50.0);
        trk.setName("TEST TRACK");

        // Submit Track
        String uid = tms_ctxt.updateCXTrack(trk, cx,"FILTER");
        // Verify Submit
        if (uid != null) System.out.println("uid = " + uid);
    }
}
```



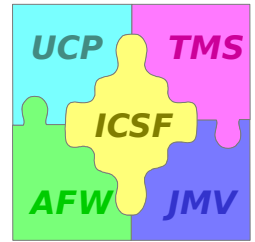
Garlic Fries Tab Merging



- ❑ *Tab Merging*
 - *Developers can implement either:*
 - `IAfwSpecifyPropertyPage` (extends `IFL`)
 - `IIflSpecifyPropertyPage`
- ❑ *Tab Merging Interface*
 - `public Object[] getPages();`
- ❑ *Usage:*
 - *The return value from this interface should contain an array of `IIflPropertyPage` objects.*



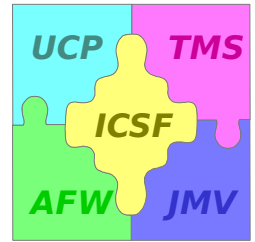
Garlic Fries Menu Merging



- ❑ *Menu Merging*
 - *Developers can use either:*
 - *IAfwSpecifyMenuItem*
 - ***ChartDBInstaller***
- ❑ *Menu Merging Interface:*
 - `public JMenuItem[] getMenuItems();`
- ❑ *Usage:*
 - *The return value from this interface should contain an array of Swing JMenuItem objects.*



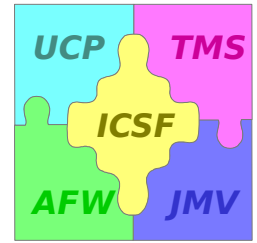
Garlic Fries Link Merging



- ❑ *Link Merging*
 - *New Functionality*
 - *IAfwSpecifyLink*
 - *Allows developers to add their URLs to a track's URL list.*
- ❑ *Link Merging Interface:*
 - *`public URL[] getLinks();`*
- ❑ *Usage:*
 - *The return value from this interface should contain an array of standard Java URL objects.*

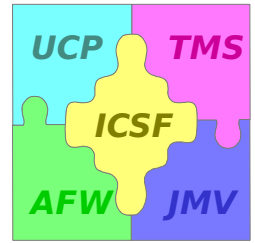


Merging Java Examples





Merging Java Examples



- ❑ *CXTrackProducer*
- ❑ *CXTrack*
- ❑ *CXTestPanel*